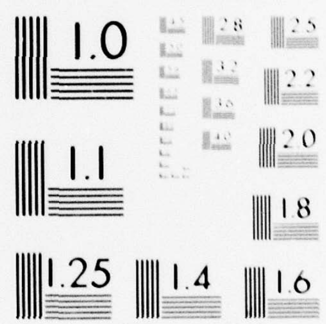
AD-A079 815 CARNEGIE-MELLON UNIV PITTSBURGH PA MANAGEMENT SCIENC--ETC F/G 12/1
AN EQUIVALENCE BETWEEN TWO ALGORITHMS FOR QUADRATIC PROGRAMMING--ETC(U)
DEC 79 J PANG
UNCLASSIFIED MSRR-452 N00014-75-C-0621

1 OF 1
ADA
079815

END
DATE
FILMED
2-80
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

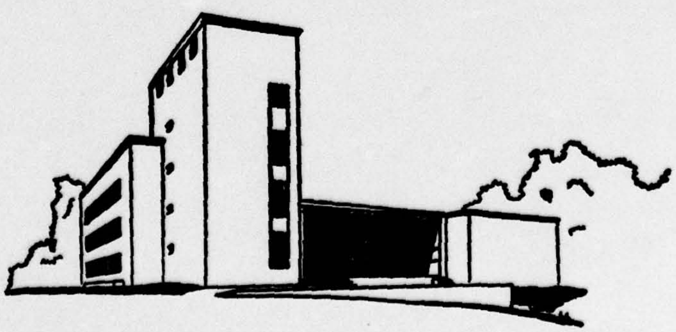
1 LEVEL II

Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER



(1) LEVEL II [redacted]

Management Sciences Research Report No. 452

AN EQUIVALENCE BETWEEN TWO ALGORITHMS FOR QUADRATIC PROGRAMMING

Jong-Shi Pang

See 1473 in back

December 1979

This report was prepared as part of the activities of the Management Sciences Research Group, Carnegie-Mellon University, under contract N00014-75-C-0621 NR 047-048 with the Office of Naval Research.

Management Sciences Research Group
Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

AN EQUIVALENCE BETWEEN TWO ALGORITHMS
FOR QUADRATIC PROGRAMMING

Jong-Shi Pang

ABSTRACT: In this paper, we demonstrate that the Van de Panne-Whinston symmetric simplex method when applied to a certain implicit formulation of a quadratic program generates the same sequence of primal feasible vectors as does the Von Hohenbalken simplicial decomposition algorithm specialized to the same program. Such an equivalence of the two algorithms extends earlier results for a least-distance program due to Cottle-Djang.

Key Words. Equivalence of algorithms, quadratic programming, column generation, simplicial decomposition.

1. <u>INTRODUCTION</u>. There has been an excessive number of algorithms for solving a convex quadratic program which are based on the Kuhn-Tucker optimality formulation of the program and employ simplex pivots. An undesirable feature of such a pivoting algorithm is that typically, the Kuhn-Tucker formulation contains more variables (multipliers of the constraints) than the program itself.

In two recent papers (Refs. 1 and 2), Von Hohenbalken has described a simplicial decomposition algorithm for solving a nonlinear minimization program with a pseudoconvex objective function and with a convex compact feasible set. The algorithm requires no dual variables and therefore eliminates the undesirable feature of having to operate on a formulation with extra variables. Another advantage of the algorithm when applied to linearly constrainted problems (such as quadratic programs) is that the powerful simplex method of linear programming can be employed to solve the subprograms.

This paper is concerned with the study of two algorithms for convex quadratic programming. One is the Von Hohenbalken algorithm and the other is the symmetric simplex method due to Van de Panne-Whinston (Ref. 3). It is shown that the former algorithm specialized to a convex quadratic program with bounded feasible set generates the same sequence of primal feasible vectors as does the latter applied to a certain implicit formulation of the given program. This result has two implications. First, the specialized Von Hohenbalken algorithm can thus be viewed as a pivoting algorithm. Second, the Van de Panne-Whinston algorithm may be implemented as a decomposition algorithm requiring no extra variables (the multipliers). As a remark, we mention that Von Hohenbalken's algorithm has been applied quite successfully for solving some fairly large quadratic programs arising from portfolio selection (Ref. 4).

It is important to point out that Van de Panne-Whinston algorithm when applied to the standard formulation of a quadratic program

$$\text{minimize } c^T x + \tfrac{1}{2} x^T C x \quad \text{subject to} \quad x \in X = \{x \in R^n : x \geqq 0,\ Ax = b,\ Dx \leqq d\} \tag{1}$$

may not produce the same sequence of primal feasible vectors as does Von Hohenbalken algorithm. An example (due to A. Djang) given in the Appendix will illustrate this fact.

Obviously, by using the representation of the feasible set X in terms of its extreme points and rays, the quadratic program (1) is equivalent to the one below

$$\text{minimize} \quad (P\eta + Q\xi)^T c + \tfrac{1}{2}(P\eta + Q\xi)^T C\ (P\eta + Q\xi) \tag{2}$$

$$\text{subject to} \qquad e^T \eta = 1 \quad ; \xi \text{ and } \eta \geqq 0 .$$

Here P and Q are the matrices of extreme points and rays of the set X respectively; and e is the vector of ones. Due to the fact that P and Q are known only implicitly, we shall call (2) the implicit formulation of the quadratic program (1). It is our contention that when Van de Panne-Whinston algorithm is applied to (2) with a vacuous Q, (i.e., with the feasible set of the original program (1) being bounded) it produces exactly the same sequence of primal feasible vectors as Von Hohenbalken algorithm is applied to (1). A noteworthy point here is that in order for the former algorithm to be applicable to (2), it is necessary to be able to implement the algorithm without the full knowledge of the matrices of generators. The tool employed to achieve this is the column generation technique described in (Ref. 5). Incidentally the algorithm described in the reference provides an alternative method for solving the quadratic program (1) without the use of multipliers. In the case where the matrix P is explicitly given, the matrix Q is vacuous, the matrix C is an identity and the vector c is zero, the quadratic program (2) reduces to the least-distance program studied in (Ref. 6) where the equivalence of the Van de Panne-Whinston and Von Hohenbalken algorithms has been established.

The idea of using the implicit formulation (2) to solve the quadratic program (1) has previously appeared in an unpublished paper by Sacher (Ref. 7) who proposed *a decomposition algorithm that involves* solving quadratic subprograms by Lemke's method. As we shall see both Van de Panne-Whinston and Von Hohenbalken algorithms require solving systems of linear equations and linear subprograms only.

The rest of the paper is organized as follows. In the next section, we describe a revised version of the Van de Panne-Whinston algorithm applied to solve a quadratic program of the form

$$\text{minimize} \quad q^T x + \tfrac{1}{2} x^T G x \quad \text{subject to} \quad x \geqq 0 \quad \text{and } Ax = b \,. \tag{3}$$

This version operates directly on the equality constraints without converting them into inequalities. In Section 3, we apply this revised algorithm to (2) and describe the column generation technique to show how the algorithm can actually be implemented without the explicit knowledge of the matrices P and Q. In Section 4, we establish a necessary and sufficient condition for an exchange pivot to occur in the application of the revised Van de Panne-Whinston algorithm to solve the implicit formulation (2) of the quadratic program. A consequence of this result is that the finite termination of the algorithm can be established without the assumption of nondegeneracy. Such conclusions extend those established by Cottle and Djang (Ref. 6) for the least-distance program. Finally, in the fifth and last section, we establish the above-mentioned equivalence between Van de Panne-Whinston and Von Hohenbalken algorithms.

2. THE SYMMETRIC SIMPLEX METHOD OF VAN DE PANNE - WHINSTON . We first summarize the operations of the Van de Panne-Whinston symmetric simplex method for solving a general convex quadratic program (Ref. 3). To start, obtain a primal feasible vector by means of a Phase I simplex method. With this vector, set up the initial standard tableau of the Kuhn-Tucker conditions so that no pair of corresponding primal and dual variables is simultaneously basic. Choose as a driving variable the primal nonbasic variable whose corresponding dual complement is most negative. If no such variable can be identified, stop; the current primal feasible vector is optimal. Otherwise, increase the driving variable until it is blocked by either its dual complement becoming nonnegative or by some other basic primal variable becoming nonpositive. If there is no blocking variable, stop; the given program has an unbounded objective value. If the blocking variable is the dual complement, perform a principal pivot making the driving primal variable basic and the corresponding blocking dual complement nonbasic (IN-PIVOT). This completes a major cycle and the algorithm attempts to find a new driving variable. If the blocking

variable is some other basic primal variable, then either a simple principal pivot or a double pivot is performed, depending on whether the former is possible. If the principal pivot is in fact possible, then a minor cycle is entered with the current driving variable unchanged (OUT-PIVOT). Such a minor cycle must terminate in a finite number of iterations with the driving variable becoming basic and its dual complement nonbasic. At that point, a new major cycle starts. Finally, if a double pivot is performed, then the current major cycle is completed and a new one begins (EXCHANGE-PIVOT).

Referring to the quadratic program (3), we say that an index set α is <u>basic feasible</u> if (i) the associated <u>basis matrix</u>

$$B(\alpha) = \begin{pmatrix} G_{\alpha\alpha} & (A._{\alpha})^T \\ -A._{\alpha} & 0 \end{pmatrix}$$

where $G_{\alpha\alpha}$ is the principal submatrix of G indexed by α and $A._{\alpha}$ consists of the columns of A indexed by α, is nonsingular; and (ii) the vector

$$\begin{pmatrix} x_{\alpha}^* \\ \lambda^* \end{pmatrix} = - B(\alpha)^{-1} \begin{pmatrix} q_{\alpha} \\ b \end{pmatrix}$$

satisfies $x_{\alpha}^* \geqq 0$.

If α is a basic feasible index set, the vectors $x^* = \begin{pmatrix} x_{\alpha}^* \\ 0 \end{pmatrix}$ and λ^*

satisfy the Kuhn-Tucker conditions of the program (3)

$$u = q + Gx + A^T\lambda \geq 0 \qquad , \qquad x \geq 0$$

$$0 = b - Ax \qquad , \qquad u^T x = 0$$

except possibly for the nonnegativity of the vector $u^* = q + Gx^* + A^T\lambda^*$. Note that $u^*_\alpha = 0$. The standard system corresponding to the (basic feasible) index set α is

$$\begin{pmatrix} x_\alpha \\ \lambda \end{pmatrix} = -B(\alpha)^{-1}\left[\begin{pmatrix} q_\alpha \\ b \end{pmatrix} + \begin{pmatrix} G_{\alpha\beta} \\ -A._\beta \end{pmatrix} x_\beta - \begin{pmatrix} u_\alpha \\ 0 \end{pmatrix}\right] \tag{4}$$

$$u_\beta = \left[q_\beta - (G_{\beta\alpha}\ (A._\beta)^T)B(\alpha)^{-1}\begin{pmatrix} q_\alpha \\ b \end{pmatrix}\right] + (G_{\beta\alpha}\ (A._\beta)^T)B(\alpha)^{-1}\begin{pmatrix} u_\alpha \\ 0 \end{pmatrix}$$

$$+ \left[G_{\beta\beta} - (G_{\beta\alpha}\ (A._\beta)^T)\ B(\alpha)^{-1}\begin{pmatrix} G_{\alpha\beta} \\ -A._\beta \end{pmatrix}\right] x_\beta$$

where β is the complement of α.

In what follows, we state a revised version of Van de Panne-Whinston algorithm applied to the quadratic program (3). This version operates with basic feasible index sets and keeps track of the useful ingredients only. In particular, it does not require the full knowledge of the system (4) and can thus be considered as an analog of the revised simplex method of linear programming.

Let α be a given basic feasible index set. Solve the system of linear equations for x_α^* and λ^*

$$B(\alpha)\begin{pmatrix} x_\alpha^* \\ \lambda^* \end{pmatrix} = -\begin{pmatrix} q_\alpha^* \\ b \end{pmatrix} \tag{5}$$

and compute

$$\bar{q}_\beta = q_\beta + G_{\beta\alpha} x_\alpha^* + (A._\beta)^T \lambda^* \tag{6}$$

The vector $\bar{q}_\beta$ gives the values of the currently basic dual variables. Determine an index $t \in \beta$ so that

$$\bar{q}_t = \min_{i \in \beta} (\bar{q}_i) \ . \tag{7}$$

If $\bar{q}_t \geqq 0$, stop; the program (3) is solved. Otherwise, solve the system of linear equations for f_α and h

$$B(\alpha)\begin{pmatrix} f_\alpha \\ h \end{pmatrix} = -\begin{pmatrix} G_{\alpha t} \\ -A._t \end{pmatrix} \tag{8}$$

and compute

$$\bar{G}_{tt} = G_{tt} + G_{t\alpha} f_\alpha + (A._t)^T h$$

Note that $\bar{G}_{tt}$ gives the diagonal entry corresponding to the pair of driving variable x_t and its complement u_t. If $\bar{G}_{tt} = 0$ and $f_\alpha \geqq 0$, stop; the program (3) is unbounded below. Otherwise, determine

$$\theta^1 = \min \{ -x_j^* / f_j \quad : \quad f_j < 0,\ j \in \alpha \} \tag{9}$$

$$\theta^2 = \begin{cases} -\bar{q}_t / \bar{G}_{tt} & \text{if } \bar{G}_{tt} > 0 \\ \infty & \text{if } \bar{G}_{tt} = 0 \end{cases}$$

If $\theta^2 < \theta^1$, add the index t to the set α and return to solve a new system of linear equations (5). This corresponds to an IN-PIVOT. If $\theta^2 \geqq \theta^1$, let s be a minimizing index in (9). Solve the system of linear equations for $\bar{f}_\alpha$ and $\bar{h}$

$$B(\alpha) \begin{pmatrix} \bar{f}_\alpha \\ \bar{h} \end{pmatrix} = \begin{pmatrix} e_\alpha^s \\ 0 \end{pmatrix} \tag{10}$$

where e_α^s is a unit vector with a one in component s. If $\bar{f}_s > 0$, drop the index s from the set α and return to solve a new system (5). Skip the comparison (7) and proceed directly to (8) after the solution of (5) with the same index t. This corresponds to an OUT-PIVOT. Finally, if $\bar{f}_s = 0$, replace α by $\alpha \setminus \{s\} \cup \{t\}$ and return to (5). This corresponds to an EXCHANGE-PIVOT. Note that $\bar{f}_s$ is the diagonal entry corresponding to the pair of blocking variable x_s and its complement u_s.

In practice, the system of linear equations (5, 8, 10) should best be solved adaptively by a factorization scheme which takes advantage of the change of the basis matrix $B(\alpha)$ (such as those described in (Ref. 8)).

We close this section by repeating an important fact. Namely, throughout the algorithm, each index set α is basic feasible.

3. THE IMPLICIT FORMULATION AND THE COLUMN GENERATION TECHNIQUE. In this section, we specialize the revised Van de Panne-Whinston algorithm to the implicit formulation (2) of the quadratic program (1). In this specialization, each basic feasible index set α is the disjoint union of two index sets α_1 and α_2 consisting of the indices of the basic η- and ξ-variables respectively. The associated basis matrix is then

$$B(\alpha_1, \alpha_2) = \begin{pmatrix} (P_{\alpha_1})^T CP_{\alpha_1} & (P_{\alpha_1})^T CQ_{\alpha_2} & e_{\alpha_1} \\ (Q_{\alpha_2})^T CP_{\alpha_1} & (Q_{\alpha_2})^T CQ_{\alpha_2} & 0 \\ -e_{\alpha_1}^T & 0 & 0 \end{pmatrix}$$

Throughout the algorithm, α_1 is always nonempty whereas α_2 may be empty. Initially, α_1 is a singleton and α_2 empty. Each index in $\alpha_1(\alpha_2)$ corresponds to an extreme point (ray) of the feasible set X. The corresponding feasible vector is given by $x = P_{\alpha_1}\eta_{\alpha_1} + Q_{\alpha_2}\xi_{\alpha_2}$ where $P_{\alpha_1}(Q_{\alpha_2})$ consists of the columns of the matrix P(Q) indexed by α_1 (α_2 respectively). These columns P_{α_1} and Q_{α_2} are to be stored after they are generated. As pointed out in (Ref. 5), there is a reasonable limit on the number of such columns required in each iteration of the algorithm.

With the index sets α_1 and α_2 and columns P_{α_1} and Q_{α_2} given, the system (5) may be written as

$$B(\alpha_1,\alpha_2)\begin{pmatrix}\eta^*_{\alpha_1}\\ \xi^*_{\alpha_2}\\ \rho^*\end{pmatrix} = -\begin{pmatrix}(P_{\alpha_1})^T c\\ (Q_{\alpha_2})^T c\\ 1\end{pmatrix} \tag{11}$$

The scalar ρ^* is the multiplier of the convexity constraint in (2). After the solution of the system of linear equations (11), one needs to determine the index t. From (6), we have

$$\begin{aligned}\bar{q}_\beta &= \begin{pmatrix}(P_{\beta_1})^T\\ (Q_{\beta_2})^T\end{pmatrix} c + \begin{pmatrix}(P_{\beta_1})^T\\ (Q_{\beta_2})^T\end{pmatrix} C(P_{\alpha_1}\ Q_{\alpha_2})\begin{pmatrix}\eta^*_{\alpha_1}\\ \xi^*_{\alpha_2}\end{pmatrix} + \rho^*\begin{pmatrix}e_{\beta_1}\\ 0\end{pmatrix}\\ &= \begin{pmatrix}(P_{\beta_1})^T\\ (Q_{\beta_2})^T\end{pmatrix}(c + Cx^*) + \rho^*\begin{pmatrix}e_{\beta_1}\\ 0\end{pmatrix}\end{aligned}$$

where β_1 and β_2 are the complements of α_1 and α_2 respectively and

$$x^* = P_{\alpha_1}\eta^*_{\alpha_1} + Q_{\alpha_2}\xi^*_{\alpha_2} . \tag{12}$$

By (11), it follows that

$$\begin{pmatrix}(P_{\alpha_1})^T\\ (Q_{\alpha_2})^T\end{pmatrix}(c + Cx^*) + \rho^*\begin{pmatrix}e_{\alpha_1}\\ 0\end{pmatrix} = 0 .$$

Therefore $\bar{q}_\beta \geqq 0$ if and only if

$$x^T(c + Cx^*) + \rho^* \geqq 0 \qquad \text{for all } x \in X .$$

Hence to determine the desired index t, one may solve the linear program

$$\text{minimize} \quad L(x^*) = x^T(c + Cx^*) \qquad \text{subject to } x \in X . \tag{13}$$

Either this program has a finite optimum $\bar{L}(x^*)$ or it is unbounded below. In the second case, an extreme ray Q_t with $t \notin \alpha_2$ is obtained. In the first case, if $\bar{L}(x^*) + \rho^* \geqq 0$, then the program (2) (and thus (1)) is solved. Otherwise an extreme vector P_t with $t \notin \alpha_1$ is obtained. In either case, if the program (2) is not solved yet, an index t and a corresponding vector P_t or Q_t are obtained such that $\bar{q}_t = \min_{i \in \beta} (\bar{q}_i)$ is negative. Note that $\bar{q}_t = \bar{L}(x^*) + \rho^*$ where $\bar{L}(x^*)$ denotes the final objective value of the linear program (13). With the index t determined, the rest of the major cycle can be completed without difficulty.

To summarize, we present below a detailed description of the revised Van de Panne-Whinston algorithm specialized to the implicit formulation (2) of the quadratic program (1).

<u>Step 0</u> (Initialization) Solve the linear program

$$\text{minimize} \quad c^T x \qquad \text{subject to } x \in X .$$

If this program is infeasible, stop; so is the quadratic program (1). Otherwise let P_1 be an extreme point feasible vector. Set $\alpha_1 = \{1\}$ and $\alpha_2 = \emptyset$. (See Remark 1.)

<u>Step 1</u> (Major Cycle) Solve the system of linear equations (11) for $\eta^*_{\alpha_1}$, $\xi^*_{\alpha_2}$ and ρ^*. Define the vector x^* by (12) and solve the linear program (13). (See Remark 2.)

<u>Step 2</u> (Termination Test) Does the final objective value $\bar{L}(x^*)$ of the linear program (13) satisfy $\bar{L}(x^*) + \rho^* \geqq 0$? If yes, stop; the current x^* is a desired optimum solution to the quadratic program (1). Otherwise, continue.

<u>Step 3</u> (Minor Cycle) Let P_t (or Q_t) be the extreme point (ray) solution obtained at the termination of the linear program (13). Solve the system of linear equations for f_{α_1}, g_{α_2} and h:

$$B(\alpha_1, \alpha_2)\begin{pmatrix} f_{\alpha_1} \\ \\ g_{\alpha_2} \\ \\ h \end{pmatrix} = -\begin{pmatrix} (P_{\alpha_1})^T CP_t \\ \\ (Q_{\alpha_2})^T CP_t \\ \\ -1 \end{pmatrix} \text{ or } -\begin{pmatrix} (P_{\alpha_1})^T CQ_t \\ \\ (Q_{\alpha_2})^T CQ_t \\ \\ -1 \end{pmatrix} \tag{14}$$

and compute the diagonal entry

$$\bar{G}_{tt} = (P_t)^T CP_t + (P_t)^T C(P_{\alpha_1} f_{\alpha_1} + Q_{\alpha_2} g_{\alpha_2}) + h \tag{15a}$$

or

$$\bar{G}_{tt} = (Q_t)^T CQ_t + (Q_t)^T C(P_{\alpha_1} f_{\alpha_1} + Q_{\alpha_2} g_{\alpha_2}) + h \tag{15b}$$

Step 4 (Test for Unboundedness) If $\bar{G}_{tt} = 0$ and $\begin{pmatrix} f_{\alpha_1} \\ g_{\alpha_2} \end{pmatrix} \geqq 0$, stop; the quadratic program (1) is unbounded below. Otherwise continue.

Step 5 (Ratio Test) Determine the minimum ratios

$$\theta_1^1 = \min \{ -\eta_i^* / f_i \quad : \quad f_i < 0 \ , \ i \in \alpha_1 \}$$

$$\theta_2^1 = \min \{ -\xi_i^* / g_i \quad : \quad g_i < 0 \ , \ i \in \alpha_2 \}$$

and let

$$\theta^1 = \min \{ \theta_1^1 , \theta_2^1 \} ,$$

$$\theta^2 = \begin{cases} -(\bar{L}(x^*) + \rho^*) / \bar{G}_{tt} & \text{if } \bar{G}_{tt} > 0 \\ \infty & \text{otherwise} . \end{cases}$$

Step 6 (In-Pivot) If $\theta^2 < \theta^1$, replace α_1 (or α_2) by $\alpha_1 \cup \{ t \}$ ($\alpha_2 \cup \{ t \}$) depending on whether P_t or Q_t is obtained at Step 3. Go to Step 1. If $\theta^2 \geqq \theta^1$, continue.

Step 7 (Check Pivot) Let s be a minimizing index in θ^1. Solve the system of linear equations for $\bar{f}_{\alpha_1}$, $\bar{g}_{\alpha_2}$ and $\bar{h}$:

$$B(\alpha_1 , \alpha_2) \begin{pmatrix} \bar{f}_{\alpha_1} \\ \bar{g}_{\alpha_2} \\ \bar{h} \end{pmatrix} = \begin{pmatrix} e^s_{\alpha_1} \\ 0 \\ 0 \end{pmatrix} \text{ or } \begin{pmatrix} 0 \\ e^s_{\alpha_2} \\ 0 \end{pmatrix}$$

depending on whether $s \in \alpha_1$ or $s \in \alpha_2$. Here $e^s_{\alpha_1}$ and $e^s_{\alpha_2}$ are unit vectors with a one in component s.

Step 8 (Exchange Pivot) If $s \in \alpha_1(\alpha_2)$ and $\bar{f}_s(\bar{g}_s) = 0$, replace $\alpha_1(\alpha_2)$ by $\alpha_1(\alpha_2) \setminus \{s\} \cup \{t\}$ and go to Step 1. Otherwise continue.

Step 9 (Out-Pivot) Replace α_1(or α_2) by $\alpha_1(\alpha_2) \setminus \{s\}$. Solve the system of linear equations (11). Retain the same vector P_t or Q_t and go to Step 3.

Remarks 1. This is just one way of getting an initial extreme point feasible vector.

2. If α_1 is a singleton and α_2 is empty (as in the initialization step), then the unique solution to the system (11) is trivial to obtain. In particular, $\pi^*_{\alpha_1}$ must be equal to 1 and thus $x^* = P_{\alpha_1}$. A similar remark holds for the systems in Steps 3 and 7.

By using (14), it is easy to see that the expressions (15a) and (15b) can be simplified as:

$$\bar{G}_{tt} = (P_t + P_{\alpha_a} f_{\alpha_1} + Q_{\alpha_2} g_{\alpha_2}) \, C \, (P_t + P_{\alpha_1} f_{\alpha_1} + Q_{\alpha_2} g_{\alpha_2}) \tag{16a}$$

$$\bar{G}_{tt} = (Q_t + P_{\alpha_1} f_{\alpha_1} + Q_{\alpha_2} g_{\alpha_2}) \, C \, (Q_t + P_{\alpha_1} f_{\alpha_1} + Q_{\alpha_2} g_{\alpha_2}) \, . \tag{16b}$$

We close this section by pointing out two more remarks. First, if the feasible set X is bounded, then Step 4 of the algorithm will never happen. It can therefore be skipped. Second, if the program (1) is strictly convex or equivalently, if the matrix C is positive definite, then by (16) and the definition of the index t, it can be easily shown that the entry $\bar{G}_{tt}$ is always positive. Two consequences of this result are: (i) again, Step 4 will never happen; and (ii) the algorithm will always compute an optimum solution of the program (1).

4. THE EXCHANGE PIVOTS. In (Ref. 6) it is shown that for the least-distance program, the exchange pivot (Step 8) is never performed because the pivot entry ($\bar{f}_s$) is always positive. A consequence of this result is that the finite termination of the algorithm can be established without the assumption of nondegeneracy, i.e., the assumption that the basic primal variables are positive in each tableau. In this section, we extend these results.

Theorem 3.1. Consider the application of the revised Van de Panne-Whinston algorithm to the implicit formulation (2) of the convex quadratic program (1) as discussed in Section 3. Then an exchange pivot is performed if and only if an in-pivot is not performed and the current index set α_1 is the singleton $\{s\}$.

To prove this lemma, we first establish

Lemma 3.2. Let C be symmetric positive semi-definite. If $B(\alpha_1, \alpha_2)$ is nonsingular, then so is each $B(\alpha'_1, \alpha'_2)$ for any $\alpha'_1 \subseteq \alpha_1$, $\alpha'_2 \subseteq \alpha_2$ and $\alpha'_1 \neq \emptyset$.

Proof. Suppose that for some such subsets α'_1 and α'_2, $B(\alpha'_1, \alpha'_2)$ is singular. Then there exist $f_{\alpha_1'}$, $g_{\alpha_2'}$ and h such that

$$B(\alpha_1', \alpha_2')\begin{pmatrix} f_{\alpha_1'} \\ g_{\alpha_2'} \\ h \end{pmatrix} = 0 .$$

It is easy to deduce that

$$(P_{\alpha_1'} f_{\alpha_1'} + Q_{\alpha_2'} g_{\alpha_2'})^T C (P_{\alpha_1'} f_{\alpha_1'} + Q_{\alpha_2'} g_{\alpha_2'}) = 0 .$$

By the given assumption on the matrix C, it follows that

$$C(P_{\alpha_1'} f_{\alpha_1'} + Q_{\alpha_2'} g_{\alpha_2'}) = 0 .$$

Thus h = 0. By defining vectors $\tilde{f}_{\alpha_1} = \begin{pmatrix} f_{\alpha_1'} \\ 0 \end{pmatrix}$ and $\tilde{g}_{\alpha_2} = \begin{pmatrix} g_{\alpha_2'} \\ 0 \end{pmatrix}$, it is easy to see that

$$B(\alpha_1, \alpha_2)\begin{pmatrix} \tilde{f}_{\alpha_1} \\ \tilde{g}_{\alpha_2} \\ 0 \end{pmatrix} = 0 .$$

Since $f_{\alpha_1'}$ and $g_{\alpha_2'}$ cannot both be zero, we obtain a contradiction Q.E.D.

<u>Proof of Theorem 3.1</u>. It suffices to show that if an in-pivot is not performed, then

(i) $s \in \alpha_2$ implies $\bar{g}_s > 0$; and

(ii) $s \in \alpha_1$ implies $\bar{f}_s > 0$ unless the current index set α_1 is the singleton $\{s\}$.

In fact, both of these assertions follow easily from the previous lemma. Q.E.D.

Theorem 3.1 implies that it is not necessary to execute Step 7 in the algorithm to determine if an exchange pivot is performed. In words an exchange pivot corresponds to an exchange of the extreme point P_s with another one P_t. The reason that such a pivot is never performed in the least-distance program is due to the choice of the initial extreme vector as one minimizing the given objective function. Such a choice is clearly impossible in the present situation because one does not know all the extreme points in advance. An exchange pivot can thus be thought of as a search for such an extreme point.

Corollary 3.3. No nondegeneracy assumption is needed for the finite termination of the algorithm.

Proof. In fact, as pointed out in (Ref. 6), a degenerate (i.e., zero) basic primal variable in any tableau may be dropped from the basis by an out-pivot. So if a tableau contains such variables, after several such out-pivots, either of two cases will arise: (i) the cardinality of α_1 is 1, or (ii) the cardinality of α_1 is greater than 1 and the ratio test (Step 5) indicates that an in-pivot should be performed. In case (i), it is easy to see that $\eta^*_{\alpha_1} = 1$ and the tableau is nondegenerate. In case (ii), the tableau must be nondegenerate in order for the in-pivot to happen. Q.E.D.

5. THE EQUIVALENCE. As Von Hahenbalken algorithm applies only to programs with compact feasible sets, we assume throughout the rest of this paper, that the feasible set X of the program (1) is bounded. In what follows, we restate the algorithm as described in (Ref. 1). Let $f(x) = c^T x + \frac{1}{2} x^T Cx$ denote the objective function, and grad denote the gradient.

Step 0 (Initialization) Solve the linear program

$$\text{minimize } x^T \text{ grad } f(0) \qquad \text{subject to } x \in X .$$

If this program is infeasible, stop; so is the quadratic program (1). Otherwise, let $\hat{x}^1$ be an extreme point of X. For t = 0, set

$$x^{t+1} = \hat{x}^1 \text{ and } B^{t+1} = \{ \hat{x}^1 \} .$$

Step 1 (Major Cycle) Set $x^t = x^{t+1}$, $B^t = B^{t+1}$ and let S^t and M^t be the simplex and the affine manifold generated by B^t. Use linear programming to locate the extreme point $\hat{x}^k$ that solves

$$\text{minimize} \quad x^T \text{ grad } f(x^t) \quad \text{subject to } x \in X \tag{17}$$

Step 2 (Termination Test) Is $(\hat{x}^k - x^t)^T$ grad $f(x^t)$ equal to zero? If yes, stop; the vector x^t is an optimal solution to the quadratic program (1). Otherwise augment the basis B^t by $\hat{x}^k$ to form the new affine basis $B = \{ \hat{x}^1 ,\dots,\hat{x}^{k-1} ,\hat{x}^k \}$.

Step 3 (Minor Cycle) Attempt to find a minimizer of f on the manifold M generated by B. If f possesses a minimizer $\bar{x}$ on M, go to Step 6. Otherwise, find its minimizer on M′ where M′ is the manifold through $\hat{x}^k$

and parallel to $M \cap M^t$. The barycentric representation of x' with respect to the basis B is $x' = Bw'$ with $w'_k > 0$ and at least one $w'_{i \neq k} \leqq 0$. Go to Step 7.

<u>Step 6</u> (In-Pivot) The barycentric representation of the minimizer $\bar{x}$ is $\bar{x} = B\bar{w}$ with $\bar{w}_k > 0$. If $\bar{w}_i > 0$ for all i, set $x^{t+1} = \bar{x}$, $B^{t+1} = B$ and go to Step 1. Otherwise continue.

<u>Step 7</u>. Intersect the line segment $\overline{x^t x'}$ or the segment $\overline{x^t \bar{x}}$ with the boundary of S, the simplex generated by B; the intersection point $x^r = Bw^r$ will have $w^r_i \geqq 0$ for all i with $w^r_k > 0$ and at least one $w^r_{i \neq k} = 0$. Let s be an index such that $w^r_s = 0$. Set $B^r = B \setminus \{ s \}$ and let S^r be the simplex generated by B^r.

<u>Step 8</u> (Exchange-Pivot) If S^r is zero-dimensional, set $x^{t+1} = x^r$, $B^{t+1} = B^r$ and to to Step 1. Otherwise continue.

<u>Step 9</u> (Out-Pivot) Set $x^t = x^r$, $B = B^r$, $M = M^r$, $S = S^r$ and go to step 3.

<u>Theorem 4.1</u>. The Van de Panne-Whinston algorithm stated in Section 3 and Von Hohenbalken algorithm stated above generate the same sequence of primal feasible vectors in the quadratic program (1).

<u>Proof</u>. To prove the theorem, we consider both algorithms entering a major cycle. By induction on the number of major cycles, we may assume that the affine basis B^t (in Von Hohenbalken algorithm) is the same as P_{α_1} (in Van de Panne-Whinston algorithm). The vector x^t is given by

$x^t = P_{\alpha_1} \pi^*_{\alpha_1} = x^*$ with x^* being the feasible vector corresponding to the index set α_1. Under this identification, it is immediately clear that the linear program (17) is precisely the one (13). The termination test (Step 2) in Von Hohenbalken algorithm is to check if the vector x^t is an optimal solution to the linear program (17). By using the equation (11) to obtain the following explicit expression for ρ^*

$$\rho^* = -(x^*)^T (c + Cx^*)$$

it is easy to see that the corresponding step in Van de Panne-Whinston algorithm is doing precisely the same thing.

After Step 2, the two algorithms start to operate somewhat differently. To establish the theorem, it suffices to show that at the completion of the major cycle, both algorithms generate the same affine basis and feasible vector. To achieve this, we use the next two lemmas.

Lemma 4.2. Suppose that both algorithms enter the minor cycle with the same basis and extreme point from Step 2. Then an in-pivot occurs in one algorithm if and only if it occurs in the other. Moreover, the feasible vectors obtained after such a pivot step are the same.

Proof. Let $B = B^r \cup \{\hat{x}^k\}$ be the affine basis in Von Hohenbalken algorithm with $B^r = P_{\alpha_1}$ being the corresponding basis in Van de Panne-Whinston algorithm. Notice that $\hat{x}^k = P_t$.

An in-pivot occurs in the former algorithm if and only if f has a minimizer $\bar{x} = B\bar{w}$ on the affine manifold M generated by B and $\bar{w} > 0$. This occurs if and only if the system

$$(P_{\alpha_1 \cup \{t\}})^T (c + CP_{\alpha_1 \cup \{t\}} w_{\alpha_1 \cup \{t\}}) + \theta e_{\alpha_1 \cup \{t\}} = 0$$

$$(e_{\alpha_1 \cup \{t\}})^T w_{\alpha_1 \cup \{t\}} = 1$$

has a solution $(\bar{w}_{\alpha_1 \cup \{t\}}, \theta)$ with $\bar{w}_{\alpha_1 \cup \{t\}} > 0$. Since the matrix

$$B(\alpha_1) = \begin{pmatrix} (P_{\alpha_1})^T CP_{\alpha_1} & e_{\alpha_1} \\ -e^T_{\alpha_1} & 0 \end{pmatrix}$$

is nonsingular, we may write the latter system as

$$\begin{pmatrix} w_{\alpha_1} \\ \theta \end{pmatrix} = \begin{pmatrix} \eta^*_{\alpha_1} \\ \rho^* \end{pmatrix} + \begin{pmatrix} f_{\alpha_1} \\ h \end{pmatrix} w_t \tag{18a}$$

$$0 = \bar{L}(x^*) + \rho^* + \bar{G}_{tt} W_t \tag{18b}$$

where $\begin{pmatrix} \eta^*_{\alpha_1} \\ \rho^* \end{pmatrix}$ and $\begin{pmatrix} f_{\alpha_1} \\ h \end{pmatrix}$ are given in (11) and (14) respectively and $\bar{G}_{tt}$ in (16a). (Recall that the matrix Q is vacuous.) Observe that the system (18) is precisely the relevant portion in the standard tableau with respect to the basic index set α_1 (cf. (4)). Since $\bar{L}(x^*) + \rho^*$ is negative (the algorithm is not terminated yet) it is obvious that the system (18) has a solution with $w_{\alpha_1 \cup \{t\}}$ positive if and only if $\theta^2 < \theta^1$ in the ratio test (Step 5 of Van de Panne-Whinston algorithm) or equivalently, an in-pivot occurs in the latter algorithm. Moreover, if such a pivot is in fact performed, it follows that both algorithms generate the same feasible vector given by $P_{\alpha_1 \cup \{t\}} \bar{w}_{\alpha_1 \cup \{t\}}$ where $\bar{w}_{\alpha_1 \cup \{t\}}$ is the unique (positive) solution to (18). This proves the lemma.

<u>Lemma 4.3</u>. Suppose that the affine basis $B^r = P_{\alpha_1}$ contains more than one vector. Then the same conclusion in Lemma 4.2 holds for an out-pivot.

<u>Proof</u>. By Lemma 4.2 and the fact that an exchange-pivot (in both algorithms) is performed if and only if an in-pivot is not performed and the current affine basis B^r (i.e., P_{α_1}) is a singleton, it suffices to verify that the feasible vectors obtained after an out-pivot is performed are the same. To prove this, observe that if f does not have a minimizer $\bar{x}$ on the manifold M generated by $B = B^r \cup \{\hat{x}^k\}$, then Step 3 of the Von Hohenbalken algorithm attempts to find a minimizer x' of f on the manifold M'. It is easy to see that the vector x' so obtained is given by $P_{\alpha_1} w'_{\alpha_1} + P_t$ where w'_{α_1} is the solution to (18a) with $w_t = 1$. Consequently, in either case, the vector x^r obtained in Step 7 of Von Hohenbalken algorithm is given by

$$x^r = \theta^* P_{\alpha_1} w_{\alpha_1} + (1-\theta^*)\, P_{\alpha_1} \eta^*_{\alpha_1} + \theta^* P_t w_t \quad \text{for some} \quad 0 \leq \theta^* \leq 1 .$$

The required θ^* is chosen to be the largest value of θ for which $w^r_{\alpha_1} = \theta w_{\alpha_1} + (1-\theta)\, \eta^*_{\alpha_1} \geq 0$. It is obviously equal to

$$\theta^* = \min \{ -\eta^*_i / (w_i - \eta^*_i) \;:\; i \in \alpha_1 \;,\; w_i < \eta^*_i \} .$$

From (18a) it follows that $\theta^* w_t = \theta^1$ where θ^1 is the minimum ratio obtained in the ratio test (Step 5) of Van de Panne-Whinston algorithm. Thus, the vector x^r is equal to

$$x^r = P_{\alpha_1} [\eta^*_{\alpha_1} + f_{\alpha_1} \theta^1] + P_t\, \theta^1$$

which is precisely the one generated by Van de Panne-Whinston algorithm. This proves the lemma. Q.E.D.

Combining these two lemmas, we obtain the desired conclusion in Theorem 4.1 readily. Finally, we point out that the two algorithms differ only in the way each minor cycle is carried out. Basically, they are the same algorithm.

6. APPENDIX. The example below (due to Professor A. Djang of the University of Kansas) shows that the Van de Panne-Whinston algorithm applied to the standard formulation (1) of a quadratic program may not produce the same sequence of primal feasible vectors as Von Hohenbalken algorithm.

Example. Consider the program

$$\text{minimize} \quad {}^{1}/_{2}(x_1^2 + x_2^2) - x_1 - 2x_2$$

$$\text{subject to} \quad 2x_1 + 3x_2 \leq 6 \quad , \quad x_1 + 4x_2 \leq 5 \quad ; \quad x_1 \quad , \quad x_2 \geq 0 \; .$$

Starting at the origin, the Van de Panne-Whinston algorithm generates the sequence: $(0,0)$, $(0,\frac{5}{4})$ and $({}^{19}/_{15}\,,\ {}^{14}/_{15})$. Starting at the same point the Von Hohenbalken algorithm generates the sequence: $(0,0)$, $(\frac{153}{97}\,,\ \frac{68}{97})$ and $({}^{19}/_{15}\,,\ {}^{14}/_{15})$.

REFERENCES

1. Von Hohenbalken, B., "A Finite Algorithm to Maximize Certain Pseudo-Concave Functions on Polytopes", Mathematical Programming 9 (1975) 189-206.

2. Von Hohenbalken, B., "Simplicial Decomposition in Nonlinear Programming", Mathematical Programming 13 (1977) 49-68.

3. Van de Panne, C. and Whinston, A., "The Symmetric Formulation of the Simplex Method for Quadratic Programming", Econometrica 37 (1969) 507-527.

4. Rosenberg, B. and Rudd, A., "Portfolio Optimization Algorithms: A Progress Report", Working Paper No. 42, Research Program in Finance, Graduate School of Business Administration, University of California, Berkeley (May 1976).

5. Pang, J.S., "A Column Generation Technique for the Computation of Stationary Points", MSRR 447, Graduate School of Industrial Administration, Carnegie-Mellon University (Oct. 1979).

6. Cottle, R.W. and Djang, A., "Algorithmic Equivalence in Quadratic Programming I: A Least-Distance Programming Problem". Report No. 76-26, Department of Operations Research, Stanford University (Oct. 1976).

7. Sacher, R.S., "An Algorithm For Large-Scale Convex Quadratic Programming," unpublished manuscript, Rensselaer Polytechnic Institute (Oct. 1976).

8. Gill, P.E., G. H. Golub, W. Murray and M.A. Saunders, "Methods for Modifying Matrix Factorizations," Mathematics of Computation 28 (1974) 505-535.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Technical Report No. 452
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): 6 AN EQUIVALENCE BETWEEN TWO ALGORITHMS FOR QUADRATIC PROGRAMMING.
5. TYPE OF REPORT & PERIOD COVERED: 9 Technical Report December 1979
6. PERFORMING ORG. REPORT NUMBER: MSRR #452
7. AUTHOR(s): 10 Jong-Shi/Pang
8. CONTRACT OR GRANT NUMBER(s): 15 N00014-75-C-0621
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR 047-048 12 28
11. CONTROLLING OFFICE NAME AND ADDRESS: Personnel and Training Research Programs, Office of Naval Research (Code 458), Arlington, Virginia 22217
12. REPORT DATE: 11 December 1979
13. NUMBER OF PAGES: 23
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report):

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION STATEMENT A

Approved for public release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Approved for public release; distribution unlimited.

9 Management Sciences research rept.,

18. SUPPLEMENTARY NOTES

14 MSRR-452

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Equivalence of algorithms, Quadratic programming, Column generation, Simplicial decomposition

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this paper, we demonstrate that the Van de Panne-Whinston symmetric simplex method when applied to a certain implicit formulation of a quadratic program generates the same sequence of primal feasible vectors as does the Von Hohenbalken simplicial decomposition algorithm specialized to the same program. Such an equivalence of the two algorithms extends earlier results for a least-distance program due to Cottle-Djang.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

403 426

JOB